Egyptian Informatics Journal (2016) 17, 227–237

Cairo University

Egyptian Informatics Journal

www.elsevier.com/locate/eij
www.sciencedirect.com

FULL-LENGTH ARTICLE

Enhanced bag of words using multilevel *k*-means for human activity recognition

Motasem Elshourbagy [a,b,*], Elsayed Hemayed [b], Magda Fayek [b]

[a] Physics and Engineering Math Department, Faculty of Engineering, Helwan University, Cairo, Egypt
[b] Computer Engineering Department, Faculty of Engineering, Cairo University, Cairo, Egypt

Received 24 August 2015; revised 22 November 2015; accepted 26 November 2015
Available online 18 April 2016

KEYWORDS

Multilevel *k*-means;
Human activity recognition;
Bag words

Abstract This paper aims to enhance the bag of features in order to improve the accuracy of human activity recognition. In this paper, human activity recognition process consists of four stages: local space time features detection, feature description, bag of features representation, and SVMs classification. The *k*-means step in the bag of features is enhanced by applying three levels of clustering: clustering per video, clustering per action class, and clustering for the final code book. The experimental results show that the proposed method of enhancement reduces the time and memory requirements, and enables the use of all training data in the *k*-means clustering algorithm. The evaluation of accuracy of action classification on two popular datasets (KTH and Weizmann) has been performed. In addition, the proposed method improves the human activity recognition accuracy by 5.57% on the KTH dataset using the same detector, descriptor, and classifier.

1. Introduction

In the last decade, the field of visual recognition had an outstanding progress from classifying instances of toy objects toward recognizing the classes of objects and scenes in natural images. Much of this progress has been sparked by the creation of realistic image and video datasets as well as by the new robust methods for image and video description and classification algorithms [1].

Today, the recognition of human activity from a video is an important area of computer vision research. It aims to analyze the activities a person is performing in a video. The video may contain an action or a sequence of actions of one human. Actions are human activity performed by a person that consists of a sequence of gestures, such as running, walking, and handclapping.

Human activity recognition shares common problems with object recognition in static images. Both tasks have to deal with significant intra-class variations, background clutter and occlusions. In the context of object recognition in static images, these problems are handled by the bag of features representation combined with machine learning techniques such

* Corresponding author at: Physics and Engineering Math Department, Faculty of Engineering, Helwan University, Cairo, Egypt. Tel.: +20 26333813, +20 01224030859.
E-mail address: motasemm@gmail.com (M. Elshourbagy).

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

ELSEVIER Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.eij.2015.11.002

as Support Vector Machines (SVM). The bag of features [2] used to represent objects in images is extended to spatiotemporal bag of features [3] to represent human activities in videos.

Standard human activity recognition consists of four stages. First stage is the detection of important interest points. Second stage is to describe the detected interest points using one or more of the descriptors such as Histogram of Oriented Gradients (HOG), Histogram of Optical Flow (HOF), Scale-Invariant Feature Transform (SIFT), and cuboids. Third stage is the bag of features algorithm, which encodes all the descriptors extracted from each video into a single code [3,4]. This simplifies the fourth classification stage.

Recently, many efforts have been done to improve bag of features algorithm. In this paper the bag of features representation has been improved to increase the accuracy of the human activity recognition. The results of the used approach are validated on the standard KTH benchmark [3] and Weizmann dataset [5,6]. The results show that our new approach improves and outperforms the state-of-the-art.

The rest of the paper is organized as follows: Section 2 presents the different video representations and encoding methods and their variations. In Section 3, the methods used for human activity recognition are presented. The framework used and the details of detector, and descriptors used are stated. The bag of features and the proposed enhancement are presented. Then the classification stage is stated. Section 4 presents and discusses the experimental results and the effect of different parameters of the proposed enhanced bag of features. Finally, Section 5 concludes the paper and suggests the future work.

2. Related work

The bag of features representation for images and videos has three important steps: code book generation, encoding of all the features in the image or video into single global statistic, and pooling and normalization of these statistics [7].

The basic method for this process is to make *k*-means clustering to generate the code book, followed by vector quantization and histogram of visual words (quantized local features) and was introduced by Sivic and Zisserman in 2003 [8,9]. This basic method produces hard vector quantization and hides a lot of information of the original features.

Other alternative methods have been proposed to overcome this problem such as Soft quantization [10,11] and local linear encoding [12]. These methods capture more information from the features by representing them as a combination of visual words. Fisher encoding [13,14] and super-vector encoding [15], record the difference between the features and visual words.

In general, to construct a code book from a set of input features, there are two approaches:

1-partioning the feature space into regions called visual code words.
2-using generative models that capture the probability distribution of features.

First approach can be achieved by using *k*-means clustering [16], hierarchal clustering [17], and spectral clustering [18]. Gaussian Mixture Models (GMM) is widely used for the second approach [13,14].

Another limitation of bag of features is its inability to encode any spatial information about relationship between words. To overcome this limitation, spatial arrangement of words is added to improve the bag of features. Some methods capture spatial relationship information of visual words by encoding the spatial arrangement of every visual word in an image [19]. Temporal bag of words model (TBoW) divides each video into N temporal bins and constructs a histogram for each bin representing points belonging to that particular bin [20]. Another encoding schemes use n-grams to augment bag of words with the discovered temporal events in a way that preserves the local structural information (relative word positions) in the activity [21].

A comparative study of encoding methods, applied to videos of action recognition, compares five encoding methods: vector quantization encoding, (localized) soft assignment encoding, sparse encoding, locality-constrained linear encoding and Fisher encoding. Sparse encoding and Fisher kernel encoding achieve higher recognition accuracy than other methods in most of the experiments [7].

Hierarchical two level clustering, clustering per video then for all the training videos, has been proposed before [22]. A new feature representation was proposed which captures the statistics of pairwise co-occurring local spatiotemporal features. The number of features produced for every video was too large, so it was constrained to a maximum limit F_{MAX} randomly sampled from the video. Features from each video are separately clustered, then all the training videos are processed together and all the obtained groups of features are reclustered to form a final codebook.

In this paper, we are proposing an enhancement for the basic bag of features with *k*-means clustering and vector quantization. The proposed approach improves the accuracy of the human activity recognition and it outperforms the new encoding methods stated before [7].

3. Framework and methods for human activity recognition

3.1. Framework of the current paper

In the current paper, the framework used consists of four stages (Fig. 1). First stage is the detection of Space Time Interest Point (STIP detector). Second stage is the calculation of descriptor at the space time volume surrounding the detected points. Third stage is the building of bag of visual words. The last stage is the classification of human activities using SVM [3]. The third stage has been modified to enhance the bag of visual words. This has been achieved by better code book generation. In order to have a better code book all the training descriptors are used. This is done using multilevel clustering. The following subsections examine these four stages in more details.

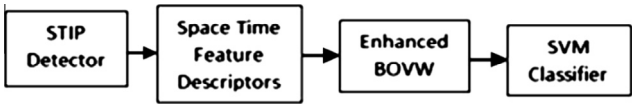

Figure 1 The framework used in the proposed method.

3.2. Space Time Interest Point detector (STIP)

Harris corner detector [23], which is used to detect corners in the spatial domain, is the most commonly used corner detector. Laptev has extended Harris corner detector to include the time as a third dimension, known as Harris 3D or Space Time Interest Point detector (STIP) [24]. STIP applies the same equations of Harris corner detector but in the x, y, and t dimensions. STIP detector is used in this paper. STIP is calculated in two steps. First calculate spatiotemporal second-moment matrix (μ) at each video point. Second calculate the function H as shown in Eqs. (1) and (2).

$$\mu(\bullet;\sigma,\tau) = g(\bullet;s\sigma,s\tau) * (\nabla L(\bullet;\sigma,\tau)(\nabla L(\bullet;\sigma,\tau))^T) \quad (1)$$

where σ is spatial scale, τ is temporal scale, g is Gaussian smoothing function, and ∇L is space-time gradients.

$$H = \det(\mu) - \mathrm{ktrace}^3(\mu) \text{ where } H > 0 \quad (2)$$

The space-time interest points are located at local maxima of the function H Eq. (2). Points are extracted at multiple scales of the scale parameters σ, τ [1]. In this study we use the original implementation available online and parameters are set as follows $k = 0.0005$, $\sigma^2 = 4, 8, 16, 32, 64, 128$, $\tau^2 = 2, 4$.

3.3. Space Time Features Descriptors

Descriptors are feature vectors describing each interesting point and its surrounding volume. The descriptors can depict shape or gradient properties in the volume such as Histogram of Oriented Gradient (HOG) [1,25]. Some descriptors depict the motion around the point using the optical flow and its histogram such as Histogram of Optical Flow (HOF) [1]. Two or more descriptors can be used together. Fusion between the features can be done at the descriptor level called early fusion or at the classifier level with a two channel classifier called late fusion. In this study, HOF and HOG descriptors were used with the KTH and Weizmann datasets, respectively. Local spatiotemporal features have demonstrated high recognition results for a number of action classes. Therefore, we use them as basic features for our approach. The selection of these specific descriptors was based on their good recognition accuracy when used with the KTH [26] and Weizmann datasets.

3.4. Enhanced bag of visual words

Bag of visual words (BOVW) is the process of encoding the video into a global descriptor in three steps. First step is to obtain a dictionary of words by clustering the descriptors obtained into words by using a clustering algorithm like k-means. Second, calculate global statistics or a histogram of these descriptors. This histogram represents the frequency of the dictionary vocabulary words in the video. Third is pooling and normalization of these statistics. k-means is usually used for clustering step. Most of the work run the k-means on a subset of the descriptors of the training videos. To limit the complexity, Laptev et al. [1,26] proposed to randomly select 100,000 features from all the training videos descriptors (input), and the number of visual words (output) was set to 4000 which has been shown to empirically give good results for a wide range of datasets.

In this study, we extend the k-means algorithm to be able to cluster all the training data and not just a small sample of it. Clustering all the training data results in enhancing the accuracy of human activity recognition. To overcome the computational complexity problem, the clustering is done in a multilevel methodology instead of clustering all input data at once. Clustering is performed for every video, for every action, and for all the actions. This can be achieved using two-level clustering or three-level clustering.

Two-level clustering is done by clustering the descriptors of each video separately in the first level into a subset (practically 10–20%) of the number of descriptors. Second level clustering is applied on the output clusters from first level to generate a single code book. This is shown in Fig. 2a.

A variant of this two-level clustering approach can be used in case of short video sequences when there are small numbers of interest points. As shown in Fig. 2b, this variant approach clusters all the videos of each action in the first layer and then the second level of clustering is applied to the output clusters from the first level to generate the final code book.

Three-level clustering is done by clustering the descriptors of each video separately in the first level into a percent of the number of descriptors in each video file. In the second level clustering, the cluster centers generated from all the videos are collected into a set for each action category and clustered each set separately into k_1 clusters. Third level of clustering is applied on the output clusters from the second level to generate a final code book of size k_2. The block diagram of the three level clustering is shown in Fig. 3.

When there are small number of interest points in each video (like the videos of Weizmann Dataset) clustering per video cannot be done. In this case the second variant of two level clustering is used instead of the first. Also three-level clustering can't be used in this case.

To increase the precision of the k-means, the algorithm is initialized 8 times and the results with the lowest error are kept [26,27].

The number of clusters or the code book size and its effect on the accuracy of the classification are studied. The effect of changing the number of clusters per action k_1 and the change of the code book size k_2 on the recognition accuracy are also inspected.

3.4.1. Computational complexity and memory requirements

Computational complexity is enhanced through multilevel k-means. Some of the symbols used in computation complexity are as follows:

N_{pv}: average number of points in the video.
N_{v}: number of videos.
N_{a}: number of actions.
$N_{\mathrm{a}i}$: number of points from the training data of action i.
$N = (N_{\mathrm{pv}} * N_{\mathrm{v}})$: total number of points from the training data.
N_1, N_2, N_3: number of points for first, second and third level.
$N_1 = N_{\mathrm{pv}}$.
$N_{\mathrm{v}i}$ = number of points in the video i.
$K_{\mathrm{v}i}$ = number of clusters for the video i.
F_{d} = feature dimension.
P: percent of points taken from each video $P = K_1/N_1$.

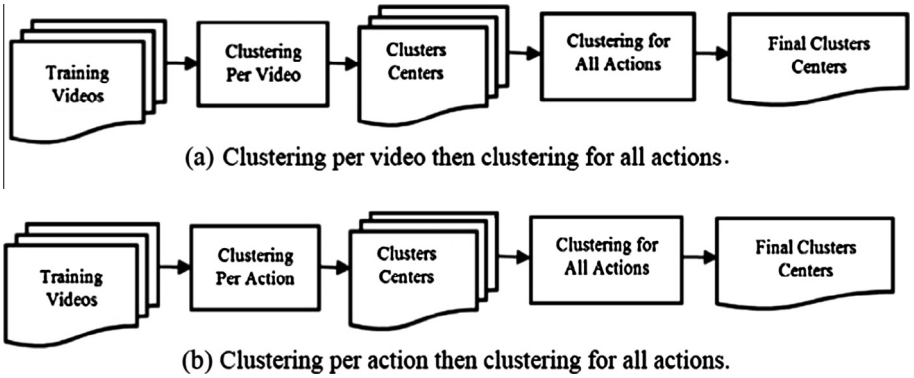

Figure 2 Two level clustering.

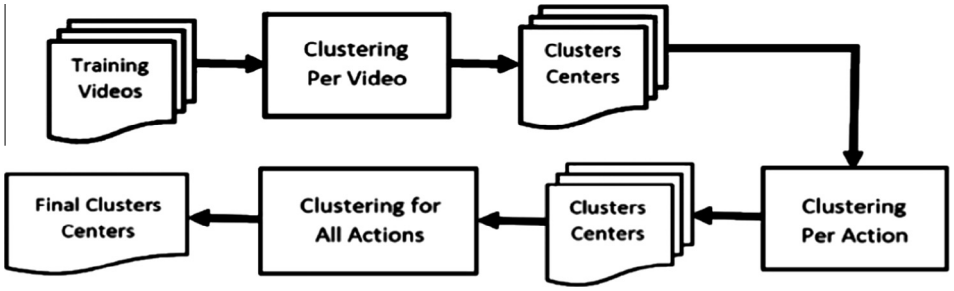

Figure 3 Three-level clustering per video, per action, and clustering for all actions.

K_1, K_2, K_3: number of clusters for first, second and third level.
K: final code book size.
O′, O″, O‴: computation complexity of one, two and three level k-means.
O_i'', O_i''': computation complexity of level number i in two and three level k-means.
Computational Complexity for One level (O′):

$$O' = N * K * F_d \tag{3}$$

which results from calculating the distance of N points of dimension F_d from K cluster centers in order to classify them.

Computational Complexity for two levels (O″):

$$O_1'' = P * F_d * \sum(N_i^2) \tag{4}$$

Computation complexity of first level, clustering per video, is the sum of the complexities of clustering all the videos of N_i points of dimension F_d into K_i cluster centers.

$$O_2'' = P * N * K * F_d \tag{5}$$

Computation complexity of second level, clustering for the final code book, results from clustering the sum of output clusters from first level ($P * N$) of dimension F_d into K cluster centers.

$$O'' = O_1'' + O_2'' \tag{6}$$

The computation complexity for two level (O″) is the sum of the complexities of the first level (O_1'') and second level (O_2''). From Eqs. (3)–(6) it follows that by taking O′ a common factor to compare it to O″ we deduce that:

$$O'' = P * \left(\sum(N_i^2)/(N * K) + 1\right) * O' \tag{7}$$

O″ is less than O′ if ($P * (\sum(N_i^2)/(N * K) + 1)$) is less than one then,

$$\sum(N_i^2) < N * K * (P^{-1} - 1) \tag{8}$$

Eq. (8) represents the condition at which O″ < O′.

Computational Complexity for three levels (O‴):

$$O''' = O_1''' + O_2''' + O_3''' \tag{9}$$

The computation complexity for three levels (O‴) is the sum of the complexities of first level (O_1'''), second level (O_2'''), and third level (O_3''').

Since $O_1'' = O_1'''$ if we prove that $O_2''' + O_3''' < O_2''$ then O‴ < O″

$$O_2''' = F_d * P * N * K_2 \tag{10}$$

The computation complexity for second level (O_2''').

$$O_3''' = F_d * N_a * K_2 * K \text{ where } K_3 = K \tag{11}$$

The computation complexity for third level (O_3''').

From Eqs. (9)–(11) it follows that the computation complexity for three levels O‴ is less than the computation complexity for two levels (O″) when:

$$K_2/K + N_a * K_2/(P * N) < 1 \text{ then } O''' < O'' \tag{12}$$

Practically $N_a * K_2/(P * N)$ is very close to zero so if $K_2 < K$ which is usually the case O‴ < O″

Memory requirements for two level k-means is less than for one level. Three-level k-means needs even less memory.

- One level

$$\mathrm{OM}' = N * K * F_{\mathrm{d}} \tag{13}$$

- Two levels

$$\mathrm{OM}'' = P * N * K * \mathrm{F_d} \tag{14}$$

$\mathrm{OM}'' = P * F_{\mathrm{d}} * \max(N_i \max^2, N * K)$

Usually $N * K$ is maximum for sparse detectors. So we have $\mathrm{OM}'' = P * \mathrm{OM}'$ which is less by $(1/P)$ times; note that P is 0.1 or 0.2.

- Three Levels

$$\mathrm{OM}''' = F_{\mathrm{d}} * K_2 * \max(p * N/N_{\mathrm{a}}, N_{\mathrm{a}} * K) \tag{15}$$

From Eqs. (14) and (15), when comparing OM''' with OM'', since $N * K \gg N/N_{\mathrm{a}} * K_2$, if $P * N > K_2 * N_{\mathrm{a}}$ then $\mathrm{OM}''' < \mathrm{OM}''$.

3.5. Support Vector Machine (SVM) classifier

Support Vector Machine (SVM) is a machine learning algorithm. It is a supervised learning algorithm that analyzes data and recognizes patterns. It is used for classification. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. SVM constructs a hyper plane or set of hyper planes in a high dimensional space, which can be used for classification. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on. In addition to performing linear classification, SVMs can efficiently perform a nonlinear classification using a kernel, implicitly mapping their inputs into high dimensional feature spaces [28]. The χ^2 Kernel is utilized in most of works for the nonlinear classification. In this paper, we use it to classify the histograms of spatial–temporal words. The χ^2 kernel is defined as follows:

$$k(x_i, x_{\mathrm{j}}) = \exp\left(-\frac{1}{A} D(x_i, x_{\mathrm{j}})\right) \tag{16}$$

where A is a scale parameter equal to the mean value of distances between all training samples [9], and it can be estimated by cross-validation. D is defined by the equation:

$$D(x_i, x_j) = \frac{1}{2} \sum_{i=1}^{p} \frac{(u_i - v_i)^2}{u_i + v_i} \tag{17}$$

where $x_i = (u_1, \ldots\ldots\ldots\ldots u_p)$ and $x_j = (v_1, \ldots\ldots\ldots\ldots v_p)$. u_i and v_i are the frequency histograms of word occurrences and p is the vocabulary size.

In case of multi-class classification the one-against-rest approach is applied and the class with the highest score is selected [28].

4. Experimental results

Four experiments have been conducted on two popular human action datasets, namely KTH and Weizmann datasets. In this section, a brief introduction for these datasets, and the experimental setup used is presented. The details and results of experiments 1 and 2 performed on KTH dataset are presented in Section 4.2. Results of experiments 3 and 4 performed on Weizmann dataset are presented in Section 4.3. Finally, the performance of the new methods has been compared with the state of the art on KTH and Weizmann datasets.

4.1. Datasets and experimental setup

KTH dataset was provided by Schuldt et al. in 2004 [3]. It consists of six action classes (boxing, hand clapping, hand waving, jogging, running and walking) and each action is performed several times by 25 subjects. The sequences were recorded in four different scenarios including indoor, outdoor, changes in clothing and variations in scale. The background is homogeneous and static in most sequences. In total, the data consist of 599 video files. The experiments follow the original experimental setup of the authors, i.e., divide the samples into test set of 9 subjects and training set of the remaining 16 subjects. The recognition results are presented in the form of average recognition rates aver all action classes.

Weizmann dataset is introduced by Gorelick et al in 2005 [5]. It consists of 10 actions (bending, jumping, jumping jack, jumping in place, running, galloping sideways, skipping, walking, one hand-waving and two hands waving). Each of these actions is performed by 9 actors resulting in 90 videos. Leave-one-person out experimental setup is used with the Weizmann dataset, where at each run 8 persons (80 videos) are used for clustering and training, and one person (10 videos) for testing. Then the average accuracy of the results is taken as the final recognition accuracy.

4.2. Experiments using KTH dataset

4.2.1. Experiment 1: Two-level clustering of KTH dataset

The first experiment is performed on KTH dataset using two levels of clustering. In the first level the descriptors from each video in the training set are clustered. In the second level all the clusters from each video in the training set are clustered again to make the dictionary. The main idea is to make two level clustering and take all the descriptors data from all the training set, so that all the descriptor features contribute to the final result.

First, the k-means algorithm is applied to each video file in the training set. The k used is a ratio of ten percent of the descriptor features extracted from the file. Second all the result clusters are used as an input to second level clustering with increasing k from 2000 to 5000 clusters by step of 500. The result code book dictionary is used to build histograms. SVM classifier is used to classify actions and the accuracy of classification is calculated on all the action classes. The effect of increasing k of the second step on the accuracy of the classifier has been calculated. The accuracy is (94.88%) at k equal 2000 clusters, and it increases with increasing k. The best accuracy (96.28%) is obtained at k equal 4000 and 4500, and then it

decreases again (Table 1). Fig. 4 shows these results with k code book size on the x-axis and the accuracy on the y-axis.

4.2.2. Experiment 2: Three-level clustering of KTH dataset

The second experiment is performed on KTH dataset using three levels of clustering. In order to further enhance the accuracy, the k-means is done on three levels, first per video, second per action and then for all actions. This is done in three steps as follows. First, the k-means algorithm is applied to each video file in the training set. The k used is a ratio of 20 percent of the number of descriptor features extracted from the file. This percent is larger than the one used before in experiment 1 to increase the data obtained from the files. Second, all the clusters obtained from videos of every action have been clustered separately, thus running six k-means algorithm one for each action using a number of clusters $k1$. Third, the resulting clusters from six actions are used as input to third level clustering for all the data to obtain the final code book. The third k-means is done with a number of clusters $k2$. The effect of changing the per action clusters $k1$ and the final clusters $k2$ on the accuracy of classification is shown in Fig. 5 and Table 2.

The results presented in Table 2 show the recognition accuracy of increasing $k1$, the per action classes, in the first column of the table and increasing $k2$, final code book size, in the first row of the table. Increasing $k2$ when $k1$ is constant usually increases the accuracy of classification to a certain limit where it starts to decrease. The row number six is representing the accuracy for $k2$ changing from 1000 to 5000 with 500 steps with $k1$ fixed at 1500. The accuracy starts at 93.5% increases to reach 95.81% at $k2 = 2000$, and then fluctuates. The highest accuracy is 96.74% at $k2 = 4000$.

Increasing $k1$ when $k2$ is constant usually increases the accuracy of classification too. The second column is representing the accuracy for $k1$ changing from 500 to 2500 with 250 steps and $k2$ fixed at 1000. The accuracy starts at 93.5% increases to reach 94% at $k1 = 1000$. The highest accuracy is 96.3% at $k1 = 1250$. Finally the best accuracy (97.7%) has been observed at $k1 = 750$, and $k2 = 4500$.

4.3. Experiments using Weizmann dataset

4.3.1. Experiment 3: One-level clustering of Weizmann dataset

This experiment is performed on Weizmann dataset using one level of clustering for all the training data. The descriptors from all the training videos are collected and clustered using k-means clustering. The code book size k is changed to find the best accuracy. The effect on accuracy of changing the code book size k is shown in Table 3.

4.3.2. Experiment 4: Two-level clustering of Weizmann dataset

This experiment is performed on Weizmann dataset using two levels of clustering. First, clustering per action has been done,

Table 1 The effect on accuracy of increasing k of the second level clustering (code book size) for experiment 1 on KTH dataset.

K	2000	2500	3000	3500	4000	4500	5000
Accuracy	94.88	95.35	94.88	95.35	**96.28**	**96.28**	95.35

Bold is best accuracy on KTH using two level clustering.

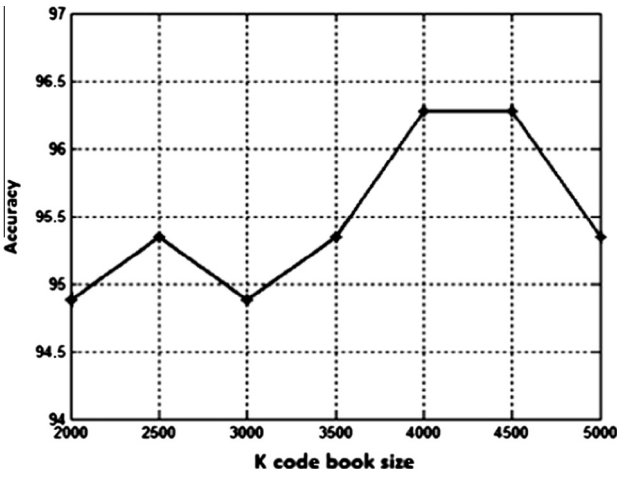

Figure 4 The effect on the accuracy of changing the codebook size in experiment 1 on KTH dataset.

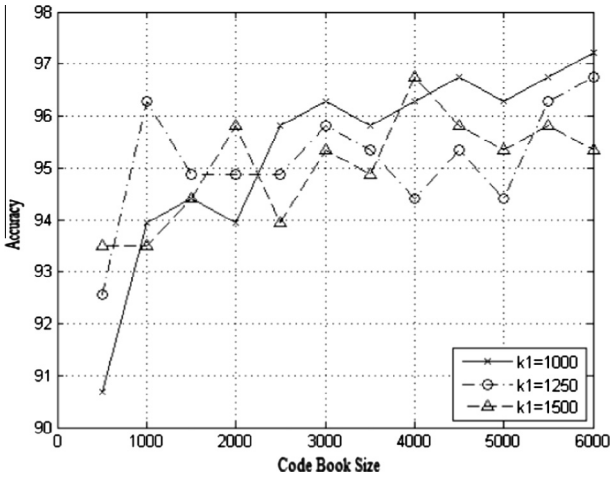

Figure 5 Sample curves for the recognition accuracy of the KTH dataset using different clustering parameters.

where the descriptors of videos from every action are clustered separately using a k-means algorithm for each action using a number of clusters $k1$. Second, the resulting clusters from six actions are used as input to second level clustering for all the data to obtain the final code book. The second k-means is done with a number of clusters $k2$. The effect of changing the per action clusters $k1$ and final clusters $k2$ on the accuracy of classification is shown in Table 4.

4.4. Comparing time of multilevel k-means

A comparison between the time of one level standard k-means, two level and three level k-means on the KTH dataset is made. The input to the k-means algorithm is a subset of 30 K, 40 K, 50 K, 100 K features randomly sampled from the training videos. The 100 K is not drawn for one level since it will take very large time. Code book size k is changed. The clustering per video is done for 10 percent and clustering per action is

Table 2 The recognition accuracy of KTH dataset using different clustering parameters: k_1 per action in the first column and k_2 final code book size in the first row.

k_1	k_2								
	1000	1500	2000	2500	3000	3500	4000	4500	5000
500	93.5	95.8	95.8	95.8	94.9	–	–	–	–
750	92.1	94.9	95.8	95.8	95.8	96.3	96.3	**97.7**	–
1000	94	94.4	94	95.8	96.3	95.8	96.3	**96.7**	96.3
1250	96.3	94.9	94.9	94.9	95.8	95.4	94.4	95.4	94.4
1500	93.5	94.4	95.8	94	95.4	94.9	**96.7**	95.8	95.4
1750	94	**96.7**	94	96.3	96.3	95.4	96.3	95.8	96.3
2000	93	94.9	95	94.9	94.9	95.8	96.3	94.9	96.3
2250	94.4	94.9	95.4	95.4	96.3	95.8	95.4	94.9	**96.7**
2500	94	95.4	95.8	94.9	94.9	95.4	95.8	**96.7**	94.9

Bold is best two values of accuracy on KTH using three level clustering.

Table 3 The recognition accuracy of Weizmann dataset using different k (code book size) in experiment 3.

K	300	400	500	600	700	800	900	1000	1100
Accuracy	94.4	95.6	96.7	95.6	94.4	93.3	**98.9**	96.7	95.6

Bold is best accuracy on Weizmann using one level clustering.

Table 4 The recognition accuracy of Weizmann dataset using different k_1 per action clusters in the first column and k_2 final code book size in the first row.

k_1	k_2								
	300	400	500	600	700	800	900	1000	1100
200	94.4	94.4	95.56	94.4	94.4	**97.8**	96.7	95.6	94.4
300	92.2	94.4	92.2	96.7	94.4	94.4	96.7	96.7	96.7
400	91.1	96.7	94.4	96.7	95.6	96.7	96.7	96.7	96.7
500	95.6	**98.9**	95.6	97.8	95.6	95.6	95.6	95.6	95.6
600	95.6	95.6	96.7	96.7	96.7	92.2	95.6	95.6	96.7
700	93.3	96.7	94.4	94.4	**97.8**	96.7	**97.8**	92.2	95.6

Bold is best two values of accuracy on Weizmann using two level clustering.

done with $k = 750$. The results are shown in Fig. 6. The time is represented on the y-axis using logarithmic scale while the code book size K is on the x-axis using linear scale. Fig. 6a compares time of one level k-means with time of two level k-means. At code book size 3000 standard k-means takes 4000 s while two level k-means takes 50 s, which means that time of standard k-means is 80 times more than two level k-means. The time of three level k-means is 40 s at the same code book size, which means that time of standard k-means is 100 times more than three level k-means.

A comparison between the time of one level standard k-means and two level k-means on the Weizmann dataset is made. All the training features are used as input and k is increased from 100 to 2500 with step of 100 for both methods. The two level k-means use $k1$ per action = 500. The results are shown in Fig. 7. At code book size less than 250 the standard k-mean takes less time than two level k-mean. At code book size more than 300 the standard k-mean takes more time than two level k-mean. As code book size increases the difference between the time taken by standard k-means and the time taken by two level k-means increases.

4.5. Comparison with the previous work

A comparison of the current best results (97.7% and 98.9%) with the previous work on KTH and Weizmann datasets is presented in Tables 5–7. Comparison was made with methods that used only local features (STIP features) and the original experimental setup on KTH [3], which have been used during this study (Tables 5 and 6).

Firstly, comparison between methods that used the same detector (Harris 3D) and same descriptor (HOF) with the current results, showed a higher accuracy of the current results (Table 5-top). The new modification introduced in the calculation of the k-means in this study could be the reason for improving the accuracy of the classification. The best result obtained by Wang et al. in 2009 [26] was 92.1% while the one achieved in this study was 97.7% (an improvement of 5.6%).

Secondly, methods that used the same detector (Harris 3D), different descriptors (HOG and HOF/HOG) and different representations were compared with the current results (Table 5-bottom). The comparison revealed that the best result on KTH dataset (94.4%) was obtained by Peng et al. in 2014 [29], and again the results achieved in this study (97.7%) outperformed their figure (an improvement of 3.3%). Peng et al. achieved their best result on KTH dataset by using GMM + FK encoding; however, the current result on KTH appeared higher than their results. This means that dividing the calculation of k-means into three levels, which is the step behind the increase in the accuracy achieved in our study, outperforms other methods for calculating the bag of visual words.

Thirdly, the comparison with methods that used different detectors and descriptors showed also that the current results were higher (Table 6). Peng et al. in 2013 [30] used recent detector and they developed new descriptor to improve accuracy on KTH. They achieved 95.6% accuracy, better than Wang et al. [27] (95%); however, the proposed method with the use of older detector and descriptor achieved a better result. Again,

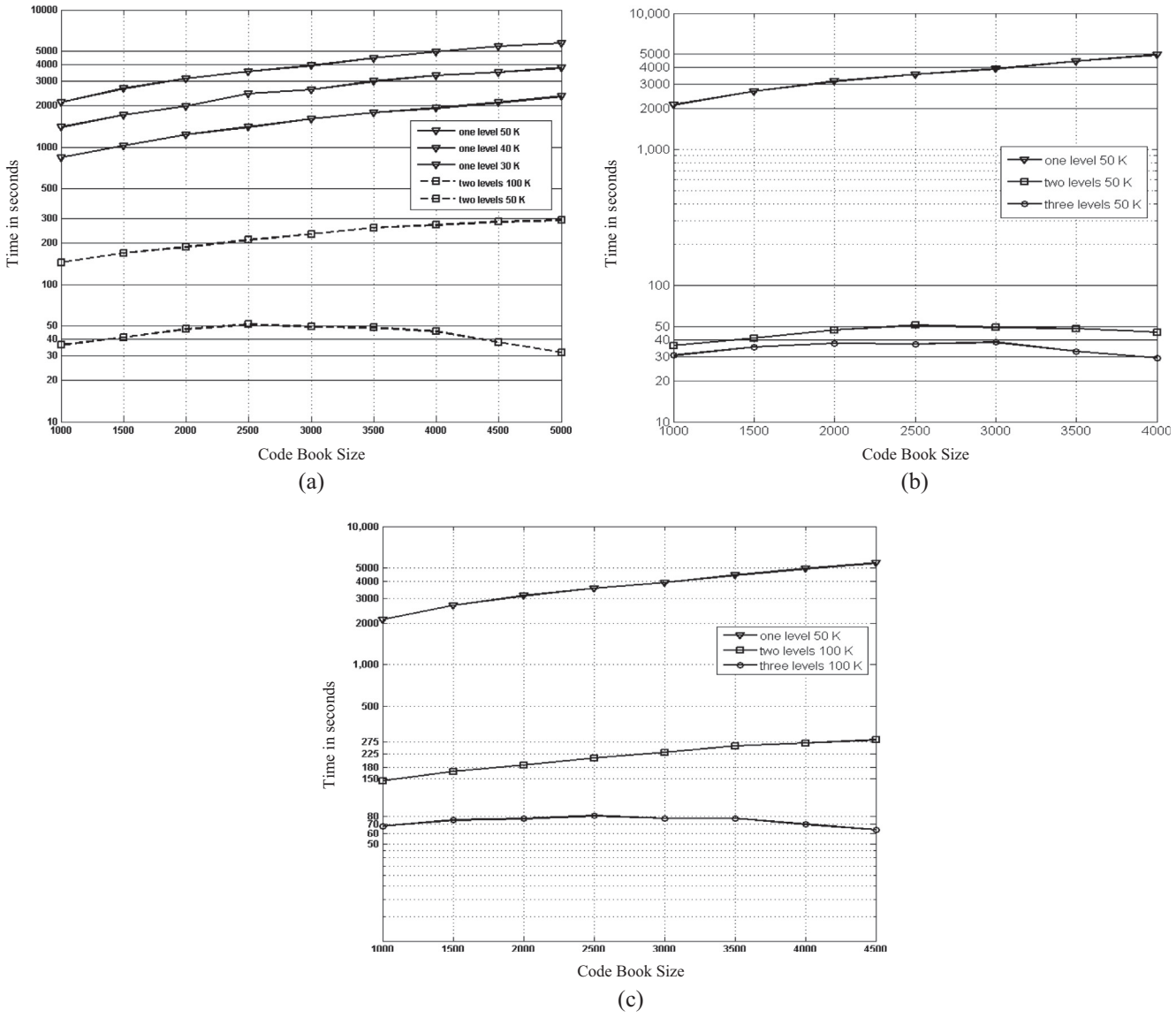

Figure 6 Comparing time for one level, two level and three level *k*-means on KTH dataset.

the achieved accuracy could be attributed to the developed modification on the *k*-means which have been introduced in our study. This modification improved the accuracy more than that obtained by the recent detector and descriptor.

Fourthly, previous work using different methods (local and global features) and different experimental setups is compared to the current results on KTH and Weizmann datasets (Table 7). Methods using different frameworks than the local features and using different additional enhancements were also compared. The developed modification outperformed all of these methods.

Klaser et al. [34] used HOG 3D descriptor, and extended integral images to integral videos for efficient 3D gradient computation, which require additional computation, however they did not obtain better results than the current study.

Bregonzio et al. [31,32] used different frameworks than the local features, easier leave one out evaluation, and global spatiotemporal distribution of the interest points was also added. This was achieved through extracting holistic features from clouds of interest points. Feature selection was applied, and frame differencing was used to extract region of interest, with extra processing. Also they made a fusion between clouds of interest points, containing complementary interest point distribution, with the conventional bag of features representation, and used a feature fusion method based on Multiple Kernel Learning. However, the developed modification in the current study achieved better results than the ones obtained by their enhancements.

Bilinski et al. [22] used Harris 3D detector, HOF and HOG descriptors and proposed a novel feature representation which captures statistics of pairwise co-occurring local spatiotemporal features. However, this method produces large number of features limited to 100,000 feature for each video and takes a lot of computation time and it achieved 96.30% accuracy using easier evaluation scheme (Leave-One-Out Cross-Validation).

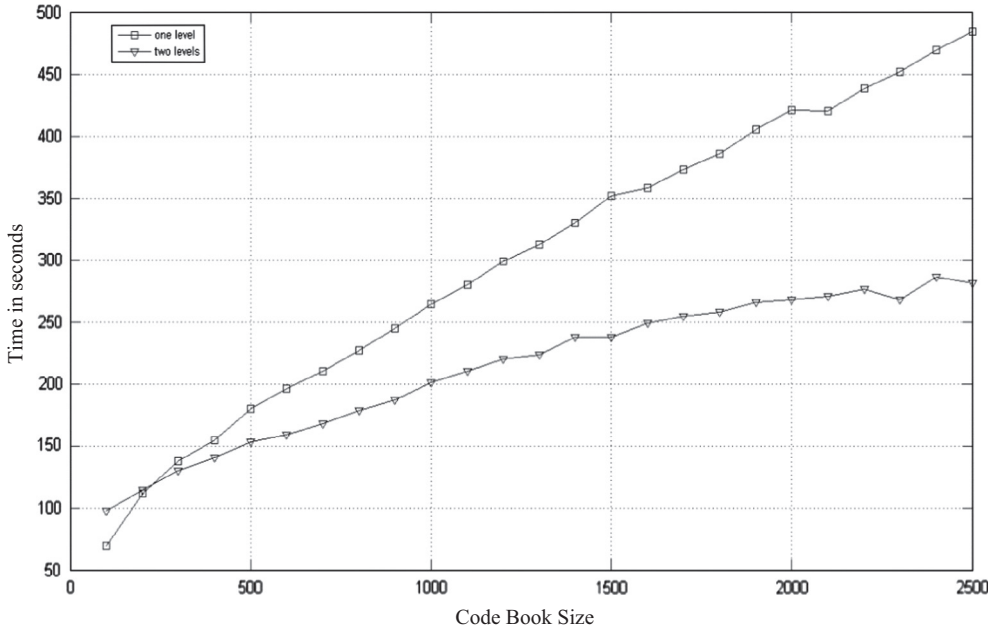

Figure 7 Comparing time for one level and two level *k*-means on Weizmann dataset.

Table 5 Comparison between current results and methods used similar detector (Harris 3D detector), similar descriptor (HOF descriptor) top and different descriptor bottom on KTH dataset.

Method	Year	Descriptor	Representation	Accuracy
Ours 3 level	2015	HOF	3 Level *k*-means + VQ	97.7
Ours 2 level	2015	HOF	2 Level *k*-means + VQ	96.3
Laptev et al. [1]	2008	HOF	BOF = *k*-means + VQ	89.7
Laptev et al.[a] [1]	2008	HOF	BOF = *k*-means + VQ	91.1
Wang et al. [26]	2009	HOF	BOF = *k*-means + VQ	92.1
Laptev et al. [1]	2008	HOG	*k*-means + VQ	81.6
Laptev et al.[a] [1]	2008	HOF/HOG	*k*-means + VQ	91.8
Wang et al. [7]	2013		*k*-means + VQ	86.1
Wang et al. [7]	2013		*k*-means + SA-all	89.8
Wang et al. [7]	2013		*k*-means + SA-*k*	88.9
Wang et al. [7]	2013		*k*-means + LLC	89.8
Wang et al. [7]	2013		*k*-means + SPC	90.7
Wang et al. [7]	2013		GMM + FK	92.1
Peng et al. [29]	2014		BOF = *k*-means + VQ	93.3
Peng et al. [29]	2014		GMM + FK	94.4

[a] Spatio-temporal grid on descriptor.

Table 6 Comparison between current results and methods used different detectors and descriptors on KTH dataset.

Method	Year	Detector	Descriptor	Accuracy
Ours 3 level	2015	Harris 3D	HOF	97.7
Ours 2 level	2015	Harris 3D	HOF	96.3
Dollar et al. [4]	2005	Cuboids	Cuboids	81.2
Wang et al. [27]	2013	DT-MB	MBH	95
Peng et al. [30]	2013	DT-MB	S-CoMBH + T-CoMBH	95.6

Table 7 Comparison between current results and previous work using different methods on KTH and Weizmann datasets.

Method	Year	KTH	Weizmann
Ours best results	2015	97.7	98.9
Bilinski et al. [22]	2012	96.30	–
Bregonzio et al. [31]	2012	94.33	96.66
Bregonzio et al. [32]	2009	93.17	96.66
Niebles et al. [33]	2008	83.3	90
Klaser et al. [34]	2008	91.4	84.3
Zhang et al. [35]	2008	91.33	92.89
Dollár et al. [4]	2005	81.16	85.2

5. Conclusion and future work

The accuracy of human activity recognition can be enhanced by the developed multilevel k-means. This modification was introduced in order to have a better code book by using all the descriptors from all training videos in the clustering stage. Achieving this in two level or three level clustering steps can be done in a reasonable time. Two-level clustering methodology was used to cluster descriptors data from each video separately and then cluster all the result clusters into a single code book. The two-level clustering enhances accuracy from 92.1% [26] to reach 96.28% using the same detector, descriptor and classifier.

A three-level clustering methodology was used to cluster descriptors data from each video separately and then cluster all the result clusters from each action into a $k1$ clusters. The cluster centers obtained from action classes were then clustered into a final single code book with $k2$ clusters. Three-level clustering methodology further enhanced the results to reach 97.7%.

Future work includes applying different distance measures in the k-means clustering such as city block distance, cosine distance instead of the Euclidian distance, which is the one used in this study.

Also, the generation of a code book for each action with different sizes and using them together with the final code book to enhance the accuracy of some confusing actions such as running and jogging is usually confused with each other. The use of the developed modification with the recent detectors and descriptors available may be explored to further enhance the accuracy of other datasets.

Acknowledgment

We thank Professor Dr. Mahmoud El Shourbagy for his great support in enhancing the results, and his help in organizing and improving the writing of the paper.

References

[1] Laptev I, Marszałek M, Schmid C, Rozenfeld B. Learning realistic human actions from movies. In: 26th IEEE conf comput vis pattern recognition, CVPR; 2008.
[2] Willamowski J, Arregui D, Csurka G, Dance C, Fan L. Categorizing nine visual classes using local appearance descriptors. ICPR 2004 workshop learning for adaptable visual systems, Cambridge, United Kingdom 22 August, 2004.
[3] Schuldt C, Laptev I, Caputo B. 2004(1321)-Recognizing human actions a local SVM approach. Pattern recognition; 2004. p. 3–7.
[4] Dollár P, Rabaud V, Cottrell G, Belongie S. Behavior recognition via sparse spatio-temporal features. In: Proc – 2nd Jt IEEE int work vis surveill perform eval track surveillance, VS-PETS; 2005. p. 65–72. http://dx.doi.org/10.1109/VSPETS.2005.1570899.
[5] Gorelick L, Blank M, Shechtman E, Irani M, Basri R. Actions as space-time shapes. ICCV IEEE, vol. 2; 2005.
[6] Gorelick L, Blank M, Shechtman E, Irani M, Basri R. Actions as space-time shapes. IEEE Trans Pattern Anal Mach Intell 2007;29:2247–53. http://dx.doi.org/10.1109/TPAMI.2007.70711.
[7] Wang X, Wang L, Qiao Y. A comparative study of encoding, pooling and normalization methods for action recognition. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics), vol. 7726 LNCS; 2013. p. 572–85. http://dx.doi.org/10.1007/978-3-642-37431-9_44.
[8] Sivic J, Zisserman A. Video Google: a text retrieval approach to object matching in videos. In: Proc ninth IEEE int conf comput vis, vol. 2; 2003. p. 1470–7. http://dx.doi.org/10.1109/ICCV.2003.1238663.
[9] Zhang J, Marszałek M, Lazebnik S, Schmid C. Local features and kernels for classification of texture and object categories: a comprehensive study. Int J Comput Vis 2007;73:213–38. http://dx.doi.org/10.1007/s11263-006-9794-4.
[10] Liu L, Wang L, Liu X. In defense of soft-assignment coding. In: Proc IEEE int conf comput vis; 2011. p. 2486–93. http://dx.doi.org/10.1109/ICCV.2011.6126534.
[11] Gemert JV, Geusebroek JM, Veenman CJ, Smeulders AWM. Kernel code-books for scene categorization. In: ECCV; 2008. p. 696–709.
[12] Wang J, Yang J, Yu K, Lv F, Huang T, Gong Y. Locality-constrained Linear Coding for image classification. In: 2010 IEEE comput soc conf comput vis pattern recognit; 2010. p. 3360–7. http://dx.doi.org/10.1109/CVPR.2010.5540018.
[13] Perronnin F, Sánchez J, Mensink T. Improving the Fisher Kernel for large-scale image classification. In: Proc 11th Eur conf comput vis part IV. Berlin, Heidelberg: Springer-Verlag; 2010. p. 143–56.
[14] Oneata D, Verbeek J, Schmid C. Action and event recognition with fisher vectors on a compact feature set. In: 2013 IEEE int conf comput vis; 2013. http://dx.doi.org/10.1109/ICCV.2013.228.
[15] Zhou X, Yu K, Zhang T, Huang TS. Image classification using super-vector coding of local image descriptors. In: Proc 11th Eur conf comput vis part V. Berlin, Heidelberg: Springer-Verlag; 2010. p. 141–54.
[16] Bishop CM. Pattern recognition and machine learning, vol. 2; 2006. http://dx.doi.org/10.1117/1.2819119.
[17] Johnson SC. Hierarchical clustering schemes. Psychometrika 1967;32:241–54. http://dx.doi.org/10.1007/BF02289588.
[18] Uw S, Ng AY, Jordan MI, Weiss Y. On spectral clustering: analysis and an algorithm. Adv Neural Inf Process Syst 2001;14:849–56.
[19] Penatti OAB, Valle E, Da S. Torres R. Encoding spatial arrangement of visual words. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics), vol. 7042 LNCS; 2011. p. 240–7. http://dx.doi.org/10.1007/978-3-642-25085-9_28.
[20] Kalra PK. Action recognition using temporal bag-of-words from depth maps. In: IAPR int conf mach vis appl. Kyoto, Japan; 2013.
[21] Bettadapura V, Schindler G, Ploetz T, Essa I. Augmenting bag-of-words: data-driven discovery of temporal and structural information for activity recognition. In: Proc IEEE comput soc conf comput vis pattern recognit; 2013. p. 2619–26. http://dx.doi.org/10.1109/CVPR.2013.338.
[22] Bilinski P, Bremond F, Local PC. Statistics of pairwise co-occurring local spatio-temporal features for human action recognition. In: 4th Int Work Video Event Categ Tagging Retr (VECTaR), conjunction with 12th Eur Conf Comput Vis (ECCV), Oct 2012, Florence, Italy; 2012. p. 311–20.

[23] Harris C, Stephens M. A combined corner and edge detector. In: Proceedings Alvey vis conf; 1988. p. 147–51. http://dx.doi.org/10.5244/C.2.23.

[24] Laptev I, Lindeberg T. Space-time interest points. In: Proc ICCV'03, Nice, Fr 2003. p. 432–9. http://dx.doi.org/10.1109/ICCV.2003.1238378.

[25] Dalal N, Triggs B. Histograms of oriented gradients for human detection. In: CVPR '05 proc 2005 IEEE comput soc conf comput vis pattern recognit, vol. 1; 2005. http://dx.doi.org/10.1109/CVPR.2005.177.

[26] Wang H, Ullah MM, Klaser A, Laptev I, Schmid C. Evaluation of local spatio-temporal features for action recognition. In: Proc Br mach vis conf; 2009. p. 124.1–124.11. http://dx.doi.org/10.5244/C.23.124.

[27] Wang H, Kläser A, Schmid C, Liu CL. Dense trajectories and motion boundary descriptors for action recognition. Int J Comput Vis 2013;103:60–79. http://dx.doi.org/10.1007/s11263-012-0594-8.

[28] Chang C, Lin C. LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol 2011;2:1–39. http://dx.doi.org/10.1145/1961189.1961199.

[29] Peng X, Wang L, Qiao Y, Peng Q. A joint evaluation of dictionary learning and feature encoding for action recognition. In: 21th Int conf pattern recognit (ICPR), Stock Sweden; 2014.

[30] Peng X. Exploring motion boundary based sampling and spatial–temporal context descriptors for action recognition. In: Br mach vis conf (BMVC), Bristol, United Kingdom; 2013. p. 1–11.

[31] Bregonzio M, Xiang T, Gong S. Fusing appearance and distribution information of interest points for action recognition. Pattern Recognit 2012;45:1220–34. http://dx.doi.org/10.1016/j.patcog.2011.08.014.

[32] Bregonzio M, Gong S, Xiang T. Recognising action as clouds of space-time interest points. In: IEEE comput soc conf comput vis pattern recognit work CVPR work; 2009. p. 1948–55. http://dx.doi.org/10.1109/CVPRW.2009.5206779.

[33] Niebles JC, Wang H, Fei-Fei L. Unsupervised learning of human action categories using spatial–temporal words. Int J Comput Vis 2008. http://dx.doi.org/10.1007/s11263-007-0122-4.

[34] Klaser A, Marszalek M, Schmid C. A spatio-temporal descriptor based on 3D-gradients. In: Br mach vis conf; 2008. http://dx.doi.org/10.5244/C.22.99.

[35] Zhang Z, Hu Y, Chan S, Chia L. Motion context: a new representation for human action recognition. Berlin Heidelb: Spinger-Verlag; 2008. p. 817–29. http://dx.doi.org/10.1007/978-3-540-88693-8_60.